ELSEVIER Linear Algebra and its Applications 309 (2000) 153–174

LINEAR ALGEBRA AND ITS APPLICATIONS

www.elsevier.com/locate/laa

QR factorization with complete pivoting and accurate computation of the SVD

Nicholas J. Higham [1]

Department of Mathematics, University of Manchester, Manchester M13 9PL, UK

Received 6 October 1998; accepted 17 September 1999

Submitted by J.L. Barlow

Abstract

A new algorithm of Demmel et al. for computing the singular value decomposition (SVD) to high relative accuracy begins by computing a rank-revealing decomposition (RRD). Demmel et al. analyse the use of Gaussian elimination with complete pivoting (GECP) for computing the RRD. We investigate the use of QR factorization with complete pivoting (that is, column pivoting together with row sorting or row pivoting) as an alternative to GECP, since this leads to a faster SVD algorithm. We derive a new componentwise backward error result for Householder QR factorization and combine it with the theory of Demmel et al. to show that high relative accuracy in the computed SVD can be expected for matrices that are diagonal scalings of a well-conditioned matrix. An a posteriori error bound is derived that gives useful estimates of the relative accuracy of the computed singular values. Numerical experiments confirm the theoretical predictions. © 2000 Elsevier Science Inc. All rights reserved.

AMS classification: 65F20; 65G05

Keywords: QR factorization; Householder matrix; Row pivoting; Row sorting; Column pivoting; Complete pivoting; Backward error analysis; Singular value decomposition; Relative accuracy; Graded matrices

1. Computing the SVD with high relative accuracy

Demmel et al. [4] consider high accuracy computation of the singular value decomposition (SVD) of $A \in \mathbb{R}^{m\times n}$. Recall that an SVD takes the form $A = U\Sigma V^{\mathrm{T}}$,

E-mail address: higham@ma.man.ac.uk (N.J. Higham).

[1] This work was supported by Engineering and Physical Sciences Research Council grant GR/L76532.

PII: S 0 0 2 4 - 3 7 9 5 (9 9) 0 0 2 3 0 - X

where U and V are orthogonal and $\Sigma = \mathrm{diag}(\sigma_i)$ contains the singular values arranged in decreasing order. Demmel et al. make use of a *rank-revealing decomposition* (RRD)

$$A = XDY^{\mathrm{T}}, \quad X \in \mathbb{R}^{m\times r}, \quad D \in \mathbb{R}^{r\times r}, \quad Y \in \mathbb{R}^{n\times r}, \quad r \leqslant \min(m,n), \tag{1.1}$$

which is defined by the properties that D is diagonal and nonsingular and X and Y are well-conditioned. The SVD itself is, of course, an RRD. The idea is to compute an RRD cheaply as the first stage of the following algorithm. (This algorithm computes only the nonzero singular values and the corresponding singular vectors.)

Algorithm SVD. Given $A \in \mathbb{R}^{m\times n}$ this algorithm computes the SVD $A = U\Sigma V^{\mathrm{T}}$.

1. Compute an RRD $A = XDY^{\mathrm{T}}$, as in (1.1).
2. Factorize $(XD)\Pi = \mathrm{QR}$ by QR factorization with column pivoting ($Q \in \mathbb{R}^{m\times r}$, $R \in \mathbb{R}^{r\times r}$).
3. Form $W = R\Pi^{\mathrm{T}}Y^{\mathrm{T}}$ (by conventional multiplication).
4. Compute the SVD $W = \overline{U}\Sigma V^{\mathrm{T}}$ using the one-sided Jacobi algorithm. (Thus $A = Q\overline{U}\Sigma V^{\mathrm{T}}$.)
5. Form $U = Q\overline{U}$.

To explain the properties of this algorithm we need to define the notion of relative accuracy of an approximate SVD, $A \approx \widetilde{U}\widetilde{\Sigma}\widetilde{V}$. This approximate SVD has relative accuracy η if, for all i,

$$|\sigma_i - \widetilde{\sigma}_i| = \mathrm{O}(\eta)\sigma_i$$

and, if σ_i is a simple singular value,

$$\sin\theta_i = \mathrm{O}\left(\frac{\eta}{\mathrm{relgap}_i}\right),$$

where θ_i denotes, in turn, the acute angle between the exact and approximate left singular vectors u_i and $\widetilde{u}_i$ and the acute angle between the exact and approximate right singular vectors v_i and $\widetilde{v}_i$, and where the relative gap is defined by

$$\mathrm{relgap}_i = \min\left(\min_{j\neq i}\frac{|\sigma_i - \sigma_j|}{\sigma_i}, 2\right).$$

Note that standard algorithms for computing the SVD, such as the QR algorithm and bisection with inverse iteration, do not provide relative accuracy; in particular, the error in the ith computed singular value is bounded relative only to the largest singular value, not to the ith: $|\sigma_i - \widehat{\sigma}_i| = \mathrm{O}(\eta)\sigma_1$.

Demmel et al. prove the following result [4, Theorems 3.1 and 3.2]. Throughout this paper the standard model of floating point arithmetic is used, with unit roundoff u (see, for example, Higham [9, Section 2.2]). The norm is the 2-norm and $\kappa(A) = \max_i \sigma_i(A)/\min_i \sigma_i(A)$.

Theorem 1.1. *Assuming that step* 1 *of Algorithm SVD is performed exactly, the computed SVD has relative accuracy* $\eta = \mathrm{O}\big(u\kappa(\overline{R})\max(\kappa(X),\kappa(Y))\big)$, *where*

$$\kappa(\overline{R}) = \min\{\kappa(DR) : D \text{ diagonal, nonsingular}\} = \mathrm{O}(\min(n2^n, n^{3/2}\kappa(X)).$$

Demmel et al. go on to analyse Gaussian elimination with complete pivoting (GECP) as the means for computing the RRD. GECP forms the factorization $PAQ = LU$, where $L \in \mathbb{R}^{m\times r}$ and $U \in \mathbb{R}^{r\times n}$, with $r = \mathrm{rank}(A)$ and where P and Q are permutation matrices. The RRD is obtained by defining $D = \mathrm{diag}(u_{ii})$ and writing $A = P^{\mathrm{T}}L \cdot D \cdot D^{-1}UQ^{\mathrm{T}} \equiv XDY^{\mathrm{T}}$. The pivoting properties ensure that PX and $Q^{\mathrm{T}}Y$ are unit lower triangular with off-diagonal elements bounded by 1; hence $\kappa(X)$ and $\kappa(Y)$ have bounds of order 2^r. In practice, $\kappa(X)$ and $\kappa(Y)$ are usually quite small, in which case GECP does indeed provide an RRD. Demmel et al. go on to show that, under certain reasonable conditions, the RRD computed by GECP is accurate enough that Theorem 1.1 remains valid, albeit with a more complicated and potentially larger bound on η.

The purpose of this paper is to investigate Householder QR factorization with column pivoting and row sorting or row pivoting as an alternative to GECP for computing the RRD in Algorithm SVD in the case of graded matrices – those that are diagonal scalings of a better conditioned matrix.

Recall that QR factorization with column pivoting produces the factorization $A\Pi = QR$, where, with $r = \mathrm{rank}(A)$, $Q \in \mathbb{R}^{m\times r}$ has orthonormal columns, $R \in \mathbb{R}^{r\times n}$ is upper trapezoidal and Π is a permutation matrix. Writing $D = \mathrm{diag}(r_{ii})$ we have $A = Q \cdot D \cdot D^{-1}R\Pi^{\mathrm{T}} \equiv XDY^{\mathrm{T}}$, with X perfectly conditioned and, in view of the inequalities

$$r_{kk}^2 \geqslant \sum_{i=k}^{j} r_{ij}^2, \quad j = k+1:n, \;\; k = 1:r \tag{1.2}$$

that the column pivoting enforces upon R, the matrix $\Pi^{\mathrm{T}}Y$ is unit lower trapezoidal with off-diagonal elements bounded by 1, and so $\kappa(Y)$ has a bound of order 2^r. Hence QR factorization with column pivoting satisfies the definition of RRD just as well as GECP.

Algorithm SVD simplifies greatly when the RRD is computed by QR factorization with column pivoting, as observed in [4]. Step 2 is unnecessary. For $X = Q$ is orthogonal and D is a diagonal matrix with elements sorted in non-increasing order, by (1.2). Therefore the QR factorization with column pivoting of XD is simply $XD \cdot I = X \cdot D$. Furthermore, step 3 is trivial since, in terms of the QR factorization determining the RRD, $W = R\Pi^{\mathrm{T}}$. We therefore have the following specialized and modified version of Algorithm SVD.

Algorithm SVD_QR. Given $A \in \mathbb{R}^{m\times n}$ this algorithm computes the SVD $A = U\Sigma V^{\mathrm{T}}$.

1. Compute the QR factorization with column pivoting $A\Pi = QR$ ($Q \in \mathbb{R}^{m\times r}$, $R \in \mathbb{R}^{r\times r}$).
2. Compute the SVD $R\Pi^{\mathrm{T}} = \overline{U}\Sigma V^{\mathrm{T}}$ using the one-sided Jacobi algorithm.
3. Form $U = Q\overline{U}$.

The main contribution of this paper is to show that, provided row sorting or row pivoting is used *in addition to* column pivoting, Householder QR factorization determines the RRD with sufficient accuracy for the theory in [4] concerning graded matrices to be applicable and to yield useful results. We refer to either of these combinations of row and column interchanges in QR factorization as *complete pivoting*. For notational simplicity we assume henceforth that $m \geqslant n$, although the overall conclusions remain true without this restriction.

In Section 2 we give a new rounding error analysis for Householder QR factorization and in Section 3 we combine the analysis with the theory of Demmel et al. Numerical experiments are given in Section 4 to confirm the practical value of the results.

2. Error analysis of Householder QR factorization

In this section we derive a new row and column-wise backward error bound for Householder QR factorization. The analysis is similar in outline to that of Cox and Higham [2] that yields a row-wise backward error bound, but it does not assume the use of column pivoting. We then derive a backward error result for complete pivoting that represents both the original matrix and the backward error matrix in a row and column scaled form, as is needed to apply the analysis of [4] for the SVD application.

First, we recall how a Householder matrix is constructed in Householder QR factorization. Let $A = A^{(1)} \in \mathbb{R}^{m\times n}$ $(m \geqslant n)$ and let $a_j^{(k)}$ denote the jth column of $A^{(k)}$, the reduced matrix at the start of the kth stage of the reduction to trapezoidal form. We form the Householder matrix

$$P_k = I - \beta_k v_k v_k^{\mathrm{T}} \in \mathbb{R}^{m\times m}, \quad \beta_k = \frac{2}{v_k^{\mathrm{T}} v_k},$$

where $v_k(1 : k-1) = 0$ and

$$v_k(k : m) = a_k^{(k)}(k : m) - \sigma_k e_1, \tag{2.1}$$

where $e_1 \in \mathbb{R}^{m-k+1}$ is the first unit vector and

$$\sigma_k = -\operatorname{sign}(a_{kk}^{(k)})\|a_k^{(k)}(k : m)\|. \tag{2.2}$$

This Householder matrix P_k has the property that $a_k^{(k+1)} = P_k a_k^{(k)}$ satisfies $a_k^{(k+1)}(k : m) = \sigma_k e_1$.

The sign of σ_k specified in (2.2) is the one recommended in most textbooks and is the one used by the QR factorization routines in LINPACK [5] and LAPACK [1]. For the other choice of sign (which can be computed in a way that avoids cancellation [9, Section 18.9]) the following lemma, on which our analysis rests, is not valid. Cox and Higham [2] have already noted the importance of the choice of sign in the Householder vector for obtaining row-wise backward error bounds.

We begin with a simple inequality.

Lemma 2.1. *The Householder vectors v_k satisfy*

$$\sqrt{2}\|a_k^{(k)}(k:m)\| \leqslant \|v_k\| \leqslant 2\|a_k^{(k)}(k:m)\|.$$

Proof. The second inequality follows from (2.1) and the first from

$$v_k^{\mathrm{T}} v_k = |2\sigma_k(\sigma_k - a_{kk}^{(k)})| \geqslant 2\sigma_k^2 = 2\|a_k^{(k)}(k:m)\|^2. \qquad \square$$

We introduce the constant

$$\tilde{\gamma}_k = \frac{cku}{1-cku},$$

in which c denotes a small integer constant whose exact value is unimportant. Hats denote computed quantities.

Rounding errors in computing the quantities β and v that determine a Householder matrix are analyzed in [9, Lemma 18.1]. By absorbing the errors in β into the vector v we can assume that β is obtained exactly. Then the computed $\widehat{v}_k \in \mathbb{R}^m$ from the kth stage of the reduction satisfies

$$\widehat{v}_k = v_k + \Delta v_k, \quad |\Delta v_k| \leqslant \tilde{\gamma}_{m-k}|v_k|, \tag{2.3}$$

where

$$P_k = I - \beta_k v_k v_k^{\mathrm{T}}$$

is the Householder matrix corresponding to the exact application of the kth stage of the algorithm to the computed matrix $\widehat{A}^{(k)}$. The following lemma is the key to the analysis.

Lemma 2.2. *Consider the computation of $\widehat{a}_j^{(k+1)} = fl(\widehat{P}_k \widehat{a}_j^{(k)})$ for $j > k$, where $\widehat{P}_k = I - \beta_k \widehat{v}_k \widehat{v}_k^{\mathrm{T}}$ and $\widehat{v}_k$ satisfies* (2.3)*. We have*

$$\widehat{a}_j^{(k+1)} = P_k \widehat{a}_j^{(k)} + f_j^{(k)}, \tag{2.4}$$

where $f_j^{(k)}(1:k-1) = 0$ and

$$|f_j^{(k)}| \leqslant u|\widehat{a}_j^{(k)}| + \tilde{\gamma}_{m-k} \frac{\|\widehat{a}_j^{(k)}(k:m)\|}{\|\widehat{a}_k^{(k)}(k:m)\|} |v_k|. \tag{2.5}$$

Proof. It is straightforward to show using standard error analysis results (see the proof of Lemma 18.2 in [9]) that (2.4) holds with $f_j^{(k)}(1:k-1) = 0$ and

$$|f_j^{(k)}| \leqslant u|\widehat{a}_j^{(k)}| + \tilde{\gamma}_{m-k}\big(|\beta_k||v_k|^{\mathrm{T}}|\widehat{a}_j^{(k)}|\big)|v_k|.$$

Now

$$\begin{aligned}\big(|\beta_k||v_k|^{\mathrm{T}}|\widehat{a}_j^{(k)}|\big)|v_k| &= 2\frac{|v_k|^{\mathrm{T}}|\widehat{a}_j^{(k)}|}{\|v_k\|^2}|v_k| \\ &\leqslant 2\frac{\|\widehat{a}_j^{(k)}(k:m)\|}{\|v_k(k:m)\|}|v_k| \\ &\leqslant \sqrt{2}\frac{\|\widehat{a}_j^{(k)}(k:m)\|}{\|\widehat{a}_k^{(k)}(k:m)\|}|v_k|,\end{aligned}$$

using Lemma 2.1; (2.5) follows. □

Now, using $P_k^2 = I$, we rewrite (2.4) as

$$\widehat{a}_j^{(k)} = P_k\widehat{a}_j^{(k+1)} - P_k f_j^{(k)}.$$

This gives

$$\begin{aligned}\widehat{a}_j^{(1)} &= P_1\widehat{a}_j^{(2)} - P_1 f_j^{(1)} \\ &= P_1(P_2\widehat{a}_j^{(3)} - P_2 f_j^{(2)}) - P_1 f_j^{(1)} \\ &\;\;\vdots \\ &= P_1P_2\cdots P_j\widehat{a}_j^{(j+1)} - P_1P_2\cdots P_j f_j^{(j)} - \cdots - P_1 f_j^{(1)}.\end{aligned}$$

Since $a_j = \widehat{a}_j^{(1)}$ and $\widehat{a}_j^{(j+1)} = \widehat{a}_j^{(n+1)}$,

$$a_j = P_1P_2\cdots P_j\widehat{a}_j^{(n+1)} - \sum_{i=1}^{j} P_1P_2\cdots P_i f_j^{(i)}. \tag{2.6}$$

Consider a general term in the sum,

$$y_i = P_1P_2\cdots P_i f_j^{(i)}, \quad i \leqslant j.$$

We have

$$\begin{aligned}y_i &= (I - \beta_1 v_1 v_1^{\mathrm{T}})P_2\cdots P_i f_j^{(i)} = P_2\cdots P_i f_j^{(i)} - \beta_1 v_1 v_1^{\mathrm{T}} P_2\cdots P_i f_j^{(i)} \\ &= (I - \beta_2 v_2 v_2^{\mathrm{T}})P_3\cdots P_i f_j^{(i)} - \beta_1 v_1 v_1^{\mathrm{T}} P_2\cdots P_i f_j^{(i)} \\ &\;\;\vdots \\ &= f_j^{(i)} - \sum_{k=1}^{i}\beta_k v_k v_k^{\mathrm{T}} P_{k+1}\cdots P_i f_j^{(i)}.\end{aligned}$$

Writing

$$z_k = \beta_k v_k v_k^{\mathrm{T}} P_{k+1}\cdots P_i f_j^{(i)} = \frac{2v_k v_k^{\mathrm{T}}}{v_k^{\mathrm{T}} v_k} P_{k+1}\cdots P_i f_j^{(i)}, \quad k \leqslant i$$

and using Lemma 2.2 we obtain

$$\begin{aligned}|z_k| &\leqslant 2|v_k|\frac{\|f_j^{(i)}\|}{\|v_k\|}\\ &\leqslant 2|v_k|\left(u\frac{\|\widehat{a}_j^{(i)}(i:m)\|}{\|v_k\|}+\tilde{\gamma}_{m-i}\frac{\|\widehat{a}_j^{(i)}(i:m)\|}{\|\widehat{a}_i^{(i)}(i:m)\|}\frac{\|v_i\|}{\|v_k\|}\right).\end{aligned}$$

Applying Lemma 2.1 leads to

$$\begin{aligned}|z_k| &\leqslant 2|v_k|\left(\frac{u}{\sqrt{2}}\frac{\|\widehat{a}_j^{(i)}(i:m)\|}{\|\widehat{a}_k^{(k)}(k:m)\|}+2\tilde{\gamma}_{m-i}\frac{\|\widehat{a}_j^{(i)}(i:m)\|}{\|v_k\|}\right)\\ &= \tilde{\gamma}_{m-i}|v_k|\left(\frac{\|\widehat{a}_j^{(i)}(i:m)\|}{\|\widehat{a}_k^{(k)}(k:m)\|}+\frac{\|\widehat{a}_j^{(i)}(i:m)\|}{\|v_k\|}\right)\\ &= \tilde{\gamma}_{m-i}\frac{\|\widehat{a}_j^{(i)}(i:m)\|}{\|\widehat{a}_k^{(k)}(k:m)\|}|v_k|.\end{aligned}$$

We conclude that

$$\begin{aligned}|y_i| &\leqslant u|\widehat{a}_j^{(i)}|+\tilde{\gamma}_{m-i}\frac{\|\widehat{a}_j^{(i)}(i:m)\|}{\|\widehat{a}_i^{(i)}(i:m)\|}|v_i|+\tilde{\gamma}_{m-i}\sum_{k=1}^{i}\frac{\|\widehat{a}_j^{(i)}(i:m)\|}{\|\widehat{a}_k^{(k)}(k:m)\|}|v_k|\\ &\leqslant u|\widehat{a}_j^{(i)}|+\tilde{\gamma}_{m-i}\sum_{k=1}^{i}\frac{\|\widehat{a}_j^{(k)}(k:m)\|}{\|\widehat{a}_k^{(k)}(k:m)\|}|v_k|,\end{aligned}$$

since $\|\widehat{a}_j^{(i)}(i:m)\| \leqslant \|\widehat{a}_j^{(k)}(k:m)\|$ for $k \leqslant i$. Note that Lemma 2.2 shows that this bound remains true if we set the first $i-1$ elements of $\widehat{a}_j^{(i)}$ to zero. Hence (2.6) can be written as

$$a_j = P_1P_2\cdots P_j\widehat{a}_j^{(n+1)}+h_j, \tag{2.7}$$

where

$$\begin{aligned}|h_j| &\leqslant u\sum_{i=1}^{j}|\widehat{a}_j^{(i)}|+\sum_{i=1}^{j}\tilde{\gamma}_{m-i}\sum_{k=1}^{i}\frac{\|\widehat{a}_j^{(k)}(k:m)\|}{\|\widehat{a}_k^{(k)}(k:m)\|}|v_k|\\ &\leqslant u\sum_{i=1}^{j}|\widehat{a}_j^{(i)}|+j\tilde{\gamma}_m\sum_{k=1}^{j}\frac{\|\widehat{a}_j^{(k)}(k:m)\|}{\|\widehat{a}_k^{(k)}(k:m)\|}|v_k|.\end{aligned} \tag{2.8}$$

But

$$P_1P_2\cdots P_j\widehat{a}_j^{(n+1)} = P_1P_2\cdots P_n\widehat{a}_j^{(n+1)} =: Q\widehat{a}_j^{(n+1)} = Q\widehat{r}_j.$$

The conclusions of the analysis are summarized in the following theorem.

Theorem 2.3. *Let $\widehat{R} \in \mathbb{R}^{m\times n}$ be the computed upper trapezoidal QR factor of $A \in \mathbb{R}^{m\times n}$ $(m \geqslant n)$ produced by the Householder QR algorithm. There exists an orthogonal $Q \in \mathbb{R}^{m\times m}$ such that*

$$A + \Delta A = Q\widehat{R},$$

where

$$|\Delta A(:, j)| \leqslant u \sum_{k=1}^{j} |\widehat{a}_j^{(k)}| + j\tilde{\gamma}_m \sum_{k=1}^{j} \frac{\|\widehat{a}_j^{(k)}(k : m)\|}{\|\widehat{a}_k^{(k)}(k : m)\|} |v_k|. \tag{2.9}$$

*In this bound the first $k-1$ elements of $\widehat{a}_j^{(k)}$ in the first summation may be set to zero. The matrix Q is given explicitly as $Q = P_1P_2\cdots P_n$, where P_k is the Householder matrix that corresponds to the exact application of the k*th *stage of the algorithm to the computed matrix produced after $k-1$ stages.*

Two existing backward error results for Householder QR factorization are implied by Theorem 2.3. First, since $\widehat{a}_j^{(k)}$ is obtained by the application of $k-1$ Householder transformations to a_j, it follows that $\|\widehat{a}_j^{(k)}\| \leqslant \|a_j\|$ (modulo roundoff). Since also $\|v_k\| \leqslant 2\|a_k^{(k)}(k : m)\|$ by Lemma 2.1, (2.9) implies $\|\Delta A(:, j)\| \leqslant j^2\tilde{\gamma}_m\|A(:, j)\|$, which is the standard column-wise backward error bound (albeit with an extra factor j).

Next, we consider QR factorization with column pivoting, in which columns are exchanged at the start of the kth stage to ensure that

$$\|a_k^{(k)}(k : m)\| = \max_{j\geqslant k} \|a_j^{(k)}(k : m)\|. \tag{2.10}$$

We will use the terminology that A is "pre-pivoted" for QR factorization with a particular interchange strategy if A is such that no interchanges are required. To apply Theorem 2.3 we assume that A is pre-pivoted for column pivoting. Using (2.9) and (2.10) we obtain (noting that $v_k(k) = a_{kk}^{(k)} - \sigma_k$ and $|\sigma_k| = \|a_k^{(k)}(k : m)\| = |a_{kk}^{(k+1)}|$)

$$|\Delta a_{ij}| \leqslant u \sum_{k=1}^{j} |\widehat{a}_{ij}^{(k)}| + j\tilde{\gamma}_m \sum_{k=1}^{j} |v_k|_i \leqslant j^2\tilde{\gamma}_m \max_{k,l} |\widehat{a}_{il}^{(k)}|, \tag{2.11}$$

which is the row-wise backward error bound of Powell and Reid [10], as obtained also by Cox and Higham [2]. The extent to which the row scaling of A is preserved in the bounds (2.9) and (2.11) is measured by the row-wise growth factor

$$\rho_{m,n} = \max_i \left\{ \frac{\max_{j,k} |a_{ij}^{(k)}|}{\max_j |a_{ij}|} \right\}. \tag{2.12}$$

This growth factor can be arbitrarily large, in general, but it can be controlled in two ways. First, we can use row pivoting: at the start of the kth stage of the factorization, after interchanging columns according to the column pivoting strategy, we interchange rows to ensure that

$$|a_{kk}^{(k)}| = \max_{i \geqslant k} |a_{ik}^{(k)}|. \tag{2.13}$$

Alternatively, we can pre-sort the rows of A so that

$$\max_j |a_{1j}| \geqslant \max_j |a_{2j}| \geqslant \cdots \geqslant \max_j |a_{mj}|. \tag{2.14}$$

Note that row interchanges before or during Householder QR factorization have no mathematical effect on the result, because they can be absorbed into the Q factor and the QR factorization is essentially unique. The effect of row interchanges is to change the intermediate numbers that arise during the factorization, and hence to alter the effects of rounding errors. Both row interchange strategies lead to a bounded growth factor, as shown by Powell and Reid [10] for row pivoting and Cox and Higham [2] for row sorting. Recall that by complete pivoting we mean column pivoting and row pivoting or row sorting.

Theorem 2.4. *For Householder QR factorization with complete pivoting applied to* $A \in \mathbb{R}^{m\times n}$,

$$\rho_{m,n} \leqslant \sqrt{m}(1+\sqrt{2})^{n-1}.$$

Although the bound of the theorem can be nearly attained, $\rho_{m,n}$ is almost always small in practice (just as for the growth factor for Gaussian elimination with partial pivoting).

The advantage of the bound (2.9) is that it is simultaneously row-wise and column-wise and so it combines the advantages of both the existing bounds. The next result shows that when complete pivoting is used and row and column scalings are factored out of A the same scalings can also be factored out of the backward error matrix. We emphasize that the diagonal matrices D_1 and D_2 in this result and in the analysis of the next section are not required by Algorithm SVD_QR but are introduced to elucidate the effect of A's scaling on the accuracy of the computed SVD.

Theorem 2.5. *Let Householder QR factorization with complete pivoting be applied to* $A \in \mathbb{R}^{m\times n}$ $(m \geqslant n)$ *and assume that* A *is pre-pivoted. Let* $\widehat{R} \in \mathbb{R}^{m\times n}$ *be the computed upper trapezoidal QR factor. For arbitrary nonsingular diagonal matrices* D_1 *and* D_2 *write* $A = D_1 B D_2$. *Let* $\rho_{m,n}$ *be the row-wise growth factor for Householder QR factorization* (*without pivoting*) *applied to* $C = D_1 B$ *and define*

$$\mu_k = \max_{j\geqslant k} \frac{\|c_j^{(k)}(k:m)\|}{\|c_k^{(k)}(k:m)\|}. \tag{2.15}$$

There exists an orthogonal $Q \in \mathbb{R}^{m\times m}$ *such that*

$$D_1(B+\Delta B)D_2 = Q\widehat{R},$$

where

$$\|\Delta B\| \leq \begin{cases} \rho_{m,n} \max_k \mu_k \, f(m,n)u\|B\| + \mathrm{O}(u^2) & \text{for row pivoting,} \\ \rho_{m,n}\psi \max_k \mu_k \, f(m,n)u\|B\| + \mathrm{O}(u^2) & \text{for row sorting,} \end{cases}$$

where

$$\psi = \max_{\substack{1\leqslant i\leqslant n \\ i\leqslant k\leqslant m}} \frac{\max_j |c_{kj}|}{\max_j |c_{ij}|} \tag{2.16}$$

and $f(m,n)$ is bounded by a low degree polynomial in m and n.

Proof. Write

$$D_i = \mathrm{diag}(d_j^{(i)}), \quad i = 1:2.$$

Let $C^{(k)} = \left(c_{ij}^{(k)}\right)$ denote the intermediate matrix at the start of the kth stage of Householder QR factorization applied to C, and let $\widetilde{v}_k$ denote the corresponding Householder vector. Since $A = CD_2$ and Householder QR factorization acts on the columns, it is clear that

$$a_j^{(k)} = c_j^{(k)} d_j^{(2)}. \tag{2.17}$$

Define $\widehat{A} =: \widehat{C}D_2$ and $\Delta A := \Delta C D_2$. Applying Theorem 2.3 to A we have $A + \Delta A = Q\widehat{R}$ and the ith component of the bound (2.9) can be written as

$$|\Delta c_{ij}|\, d_j^{(2)} \leqslant u \sum_{k=1}^{j} |\widehat{c}_{ij}^{(k)}| d_j^{(2)} + j\tilde{\gamma}_m \sum_{k=1}^{j} \frac{\|\widehat{c}_j^{(k)}(k:m)\| d_j^{(2)}}{\|\widehat{c}_k^{(k)}(k:m)\| d_k^{(2)}} |\widetilde{v}_k|_i d_k^{(2)},$$

which gives

$$|\Delta c_{ij}| \leqslant j\tilde{\gamma}_m \left(\sum_{k=1}^{j} |\widehat{c}_{ij}^{(k)}| + \sum_{k=1}^{j} \frac{\|\widehat{c}_j^{(k)}(k:m)\|}{\|\widehat{c}_k^{(k)}(k:m)\|} |\widetilde{v}_k|_i \right).$$

Now $\|\widehat{c}_j^{(k)}(k:m)\| / \|\widehat{c}_k^{(k)}(k:m)\| \leqslant \mu_k$ for $j \geqslant k$ by (2.15) and

$$|\widehat{c}_{ij}^{(k)}| \leqslant \rho_{m,n} \max_r |c_{ir}|,$$

so

$$|\Delta c_{ij}| \leqslant j\tilde{\gamma}_m \left(j\rho_{m,n} \max_r |c_{ir}| + \sum_{k=1}^{j} \mu_k |\widetilde{v}_k|_i \right).$$

(We are treating μ_k and $\rho_{m,n}$ as if they were defined in terms of computed instead of exact quantities, which does not affect the bounds, to first order.) We now bound the $|\widetilde{v}_k|_i$ term. For $i > k$ we have

$$|\widetilde{v}_k|_i = |\widehat{c}_{ik}^{(k)}| \leqslant \rho_{m,n} \max_r |c_{ir}|.$$

For $i = k$ slightly different analysis is required for row pivoting and row sorting. Note from (2.17) that since A does not require row interchanges for complete pivoting with row pivoting, neither does C. Hence for row pivoting

$$|\widetilde{v}_k|_k = |c_{kk}^{(k)}| + \|c_k^{(k)}(k:m)\| \leqslant (1+\sqrt{m-k+1})|c_{kk}^{(k)}|$$
$$\leqslant (1+\sqrt{m-k+1})\rho_{m,n}\max_r |c_{kr}|.$$

For row sorting

$$\begin{aligned}|\widetilde{v}_k|_k &= |c_{kk}^{(k)}| + \|c_k^{(k)}(k:m)\| \\ &\leqslant |c_{kk}^{(k)}| + \sqrt{m-k+1}\max_{p\geqslant k}|c_{pk}^{(k)}| \\ &\leqslant |c_{kk}^{(k)}| + \sqrt{m-k+1}\max_{p\geqslant k}\rho_{m,n}\max_r |c_{pr}| \\ &\leqslant (1+\sqrt{m-k+1})\rho_{m,n}\psi\max_r |c_{kr}|,\end{aligned}$$

where ψ is given by (2.16). (Note that, unlike for row pivoting, C may require interchanges for row sorting.)

Hence, overall,

$$|\Delta c_{ij}| \leqslant j\big(j+\max_k \mu_k(\sqrt{m}+j)\big)\tilde{\gamma}_m\rho_{m,n}\max_r |c_{ir}|,$$

where an extra factor ψ is needed for row sorting. Defining $\Delta C =: D_1\Delta B$ and using $C = D_1 B$, this bound can be written

$$d_i^{(1)}|\Delta b_{ij}| \leqslant j\big(j+\max_k \mu_k(\sqrt{m}+j)\big)\tilde{\gamma}_m\rho_{m,n}d_i^{(1)}\max_r |b_{ir}|,$$

and the result follows on cancelling the term $d_i^{(1)}$ and taking norms. □

Drmač [6, Section 2.3] obtains a similar result to Theorem 2.5 for row pivoting under the assumption that $\mu_k = 1$ for all k. The result of Theorem 2.5 is stronger for row pivoting than for row sorting, since the factor ψ can be arbitrarily large for row sorting.

The diagonal matrices D_1 and D_2 in Theorem 2.5 are arbitrary and they determine the size of the μ_k in (2.15) and the growth factor $\rho_{m,n}$. As long as these quantities are of order 1 we have a backward error bound that scales perfectly under these particular row and column scalings. The size of μ_k depends on how close C is to being pre-pivoted for column pivoting. As noted in the proof, C *is* pre-pivoted for row pivoting. Although the bound for $\rho_{m,n}$ in Theorem 2.4 is valid only for complete pivoting, it is not hard to show that for C in Theorem 2.5 the modified bounds

$$\rho_{m,n} \leqslant \begin{cases} \sqrt{m}\displaystyle\prod_{k=1}^{n-1}(1+\sqrt{2}\,\mu_k) & \text{for row pivoting,} \\ \psi\sqrt{m}\displaystyle\prod_{k=1}^{n-1}(1+\sqrt{2}\,\mu_k) & \text{for row sorting,}\end{cases}$$

hold. These bounds are weak, but they show that $\rho_{m,n}$ can be bounded in terms of the μ_k.

If we are given A and wish to choose D_1, D_2 and hence B, then two natural choices are

$$D_1 = \mathrm{diag}(\|A(i,:)\|), \quad D_2 = \mathrm{diag}(\|A(:,j)\|), \tag{2.18}$$

$$D_1 = \mathrm{diag}(\|A(i,:)\|), \quad D_2 = \mathrm{diag}(\|D_1^{-1}A(:,j)\|) \tag{2.19}$$

(where we continue to assume that A is pre-pivoted with respect to both row and column interchanges). Assuming that A has no zero rows or columns, the choice (2.19) produces a matrix B whose rows and columns are approximately equilibrated in the 2-norm; indeed, for all i and j, $\|B(:,j)\| = 1$ and $n^{-1/2} \leqslant \|B(i,:)\| \leqslant n^{1/2}$. For the choice (2.18), $\|B\|$ is unbounded and all we can say is that $\max_{i,j} |b_{ij}| \geqslant (n\|A\|)^{-1}$. This latter choice does not necessarily produce a B that is approximately row and column equilibrated: for

$$A = \begin{bmatrix} \theta & 1 \\ 1 & 1 \end{bmatrix}, \quad \theta \gg 1,$$

we have

$$B = D_1^{-1} A D_2^{-1} \approx \begin{bmatrix} \theta^{-1} & \theta^{-1} \\ \theta^{-1} & 1 \end{bmatrix} = \theta^{-1} \begin{bmatrix} 1 & 1 \\ 1 & \theta \end{bmatrix},$$

which is far from being row or column equilibrated.

The ratios μ_k are unbounded for (2.18), as is shown by the matrix

$$A = \begin{bmatrix} \theta & \theta/2 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad \theta \gg 1,$$

for which $\mu_2 \approx \theta$. For (2.19) we are not aware of such an example.

3. Application to computation of the SVD

To show that Householder QR factorization with complete pivoting can be used to compute the RRD in Algorithm SVD in place of GECP, we need to show that the RRD obtained is of sufficient accuracy. We need a result from [4, Theorem 4.1].

Theorem 3.1. *Let $A \in \mathbb{R}^{m\times n}$ $(m \geqslant n)$ be written $A = D_1 B D_2$, for arbitrary nonsingular diagonal matrices D_1 and D_2. Let $A + \Delta A = D_1(B + \Delta B)D_2$ and assume that B has an LU factorization (without pivoting) $B = LU$. Write $L = [\, L_{11}^{\mathrm{T}} \quad L_{21}^{\mathrm{T}} \,]^{\mathrm{T}}$ and define*

$$\widetilde{L} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & I_{m-n} \end{bmatrix} \in \mathbb{R}^{m\times m}, \quad L^{\#} = \begin{bmatrix} L_{11}^{-1} \\ -L_{21}L_{11}^{-1} \end{bmatrix} \in \mathbb{R}^{m\times n}.$$

Then, for all i,

$$\frac{|\sigma_i(A+\Delta A) - \sigma_i(A)|}{\max(\sigma_i(A+\Delta A), \sigma_i(A))} \leqslant \tau\big(\kappa(\widetilde{L}) + \kappa(U)\big)\|L^{\#}\|\|U^{-1}\|\|\Delta B\| + \mathrm{O}(\|\Delta B\|^2),$$

where

$$\tau = \max\{\,\tau_1, \tau_2\,\} \geqslant 1, \tag{3.1a}$$

$$\tau_1 = \max_{\substack{1\leqslant i\leqslant n\\ i\leqslant j\leqslant m}} \frac{D_1(j,j)}{D_1(i,i)}, \quad \tau_2 = \max_{1\leqslant i\leqslant j\leqslant n} \frac{D_2(j,j)}{D_2(i,i)}. \tag{3.1b}$$

A similar result can be proven for the singular vectors [4], so we conclude that the SVDs of A and $A + \Delta A = D_1(B + \Delta B)D_2$ agree to relative accuracy of order u when the following three conditions hold:

1. $\|\Delta B\|/\|B\|$ is of order u.
2. B has an LU factorization and the factors are well-conditioned or, equivalently, the leading principal submatrices of B are well-conditioned.
3. τ in (3.1) is of order 1.

We identify $A + \Delta A$ with the row and column permuted matrix whose QR factorization with complete pivoting we compute in floating point arithmetic, and we define D_1 and D_2 by (2.18) or (2.19).

We consider first the size of τ in (3.1). Row sorting ensures that $\tau_1 \leqslant \sqrt{n}$. For row pivoting, however, τ_1 is unbounded. For example, if every row of A is a multiple of the same vector then after the first step of Householder QR factorization the active submatrix is zero and so only one row interchange is carried out during the whole factorization; hence $D_1(3,3)/D_1(2,2)$ can be arbitrarily large, for example. Similarly, for a low rank matrix column pivoting may not have the opportunity to interchange all the columns and hence τ_2 is unbounded. For a less trivial example of how τ_2 can be arbitrarily large for (2.18) with column pivoting, consider the matrix

$$A = \begin{bmatrix} \theta & 0 & \theta/2 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \theta \gg 1.$$

In QR factorization with column pivoting, $\Pi = I$ and $\tau_2 = \theta/2$. However, in practice, we would expect row pivoting to roughly sort the rows by ∞-norm and column pivoting to roughly sort the columns by 2-norm, yielding small values of τ_1 and τ_2. Therefore in practice we would expect condition 3 to be satisfied for both choices of D_1 and D_2 if we use complete pivoting.

In view of Theorem 2.5, condition 1 will be satisfied[2] provided that $D_1 B$ is approximately pre-pivoted for QR factorization with column pivoting, the growth factor $\rho_{m,n}$ is not too large and, for row sorting, ψ in (2.16) is not too large.

[2] A subtlety is that we would seem to require an analogue of Theorem 2.3 in which Q is replaced by the computed product of Householder matrices, since Algorithm SVD_QR requires the explicit Q to define X in the RRD. However, in Algorithm SVD_QR there is clearly no need to form Q explicitly, and the errors associated with applying Q to $\overline{U}$ in the last step of the algorithm are covered by the analysis of step 5 of Algorithm SVD in [4].

Condition 2 is beyond our control once we have chosen D_1 and D_2. As explained in [4], the condition of submatrices of B plays an intrinsic role in the sensitivity of the SVD of D_1BD_2 to perturbations in B.

We now summarize the analysis for GECP in [4]. We can take the same definitions (2.18) or (2.19) of D_1 and D_2 (based now on the ∞-norm instead of the 2-norm) and let $A + \Delta A$ be the row and column permuted matrix whose LU factorization with complete pivoting is computed. Because GE is invariant under row and column scalings for a fixed pivot sequence (see, for example, [9, Section 9.7]), $\|\Delta B\|/\|B\|$ has the usual error bound for GE without pivoting on B [9, Theorem 9.3], and this bound will be small if condition 2 is satisfied. The quantity τ is unbounded, though again is expected to be acceptably small in practice. In [4] some additional analysis is given that bypasses Theorem 3.1 and avoids an explicit choice of D_1 and D_2; see [4, Theorem 4.2 and Corollary 4.1] for details.

The main difference between the conclusions for QR factorization with complete pivoting and those for GECP, then, is in the conditions required to guarantee that $\|\Delta B\|/\|B\|$ is of order u. The condition in the former case that the matrix D_1B is approximately pre-pivoted is replaced for GECP by the condition that the leading principal submatrices of B are well-conditioned; this latter condition is a strong one, but is already required for the RRD to determine the SVD sufficiently accurately. In summary, then the available analysis imposes weaker conditions for Algorithm SVD to work with GECP than for with QR factorization with complete pivoting.

Finally, we consider how to test, after computing the SVD, whether the desired relative accuracy was achieved. We concentrate on assessing the effect of rounding errors in computing the RRD; Theorem 1.1 describes the effect of errors in the other parts of Algorithm SVD_QR. The first possibility is to use Theorem 3.1. For GECP the main tasks are to evaluate τ and to estimate the condition numbers of the LU factors of B. The backward error matrix ΔB can be bounded from the standard backward error bound. For QR factorization with column pivoting we must explicitly compute the LU factorization of B, since it is not already available. We can evaluate the bound (2.9) for ΔA and thereby explicitly compute a bound for ΔB (or a sharper running error bound derived directly from the equations defining the algorithm; see [9, Section 3.3] for details of this general technique). The cost of evaluating (2.9) is $3(mn^2/2 - n^3/6)$ operations (the terms $\|\widehat{a}_j^{(k)}(k:m)\|$ are already available as they are needed for the column pivoting strategy).

The second way to obtain an a posteriori error bound avoids explicit use of the LU factors and does not explicitly involve D_1 and D_2. We need the following result from [4, Theorem 2.2; 7, Theorem 3.1].

Theorem 3.2. *Let $A \in \mathbb{R}^{m\times n}$ and $\widetilde{A} = (I + E)A(I + F)$ have singular values σ_i and $\widetilde{\sigma}_i$, respectively, where $E \in \mathbb{R}^{m\times m}$ and $F \in \mathbb{R}^{n\times n}$. Then*

$$|\sigma_i - \widetilde{\sigma}_i| \leqslant |\sigma_i|(\eta_E + \eta_F + \eta_E\eta_F),$$

where $\eta_E = \|E\|$ and $\eta_F = \|F\|$.

The next theorem is a generalization of Theorem 4.2 in [4]. We use the $m \times n$ identity matrix $I_{m,n} = (\delta_{ij})$.

Theorem 3.3. *Let $A \in \mathbb{R}^{m\times n}$ and $A + \Delta A = ST$ have singular values σ_i and $\widetilde{\sigma}_i$, respectively, where*

$$S = \begin{matrix} \\ n \\ m-n \end{matrix} \begin{matrix} \begin{matrix} n \end{matrix} \\ \begin{bmatrix} S_{11} \\ S_{21} \end{bmatrix} \end{matrix} \in \mathbb{R}^{m\times n}$$

and $T \in \mathbb{R}^{n\times n}$ is nonsingular. Let

$$\widetilde{S} = \begin{bmatrix} S_{11} & \widetilde{S}_{12} \\ S_{21} & \widetilde{S}_{22} \end{bmatrix} \in \mathbb{R}^{m\times m},$$

where $\widetilde{S}_{12} \in \mathbb{R}^{n\times(m-n)}$ and $\widetilde{S}_{22} \in \mathbb{R}^{(m-n)\times(m-n)}$ are arbitrary subject to $\widetilde{S}$ being nonsingular. Assume that $I_{m,n} - \widetilde{S}^{-1}\Delta A T^{-1}$ has an LU factorization without pivoting. Then

$$|\sigma_i - \widetilde{\sigma}_i| \leqslant |\sigma_i|\epsilon + \mathrm{O}(\epsilon^2),$$

where

$$\epsilon = \|\widetilde{S}\,[\,\mathrm{tril}(\widetilde{S}^{-1}\Delta A T^{-1})\ 0\,]\,\widetilde{S}^{-1}\| + \|T^{-1}\,\mathrm{triu}(\widetilde{S}^{-1}\Delta A T^{-1})T\|,$$

where tril *and* triu *denote the strictly lower trapezoidal part and the upper triangular part* (*including the diagonal*), *respectively.*

Proof. We have

$$A + \Delta A = ST \equiv \widetilde{S} I_{m,n} T.$$

Hence

$$\begin{aligned} A &= \widetilde{S} I_{m,n} T - \Delta A \\ &= \widetilde{S}(I_{m,n} - \widetilde{S}^{-1}\Delta A T^{-1})T \\ &= \widetilde{S} L U T, \end{aligned}$$

where

$$I_{m,n} - \widetilde{S}^{-1}\Delta A T^{-1} = LU = \begin{bmatrix} L_{11} & 0 \\ L_{21} & I_{m-n} \end{bmatrix} I_{m,n} U \equiv \widetilde{L} I_{m,n} U, \quad \widetilde{L} \in \mathbb{R}^{m\times m},$$

is an LU factorization without pivoting. Hence

$$\begin{aligned} A &= \widetilde{S}\widetilde{L} I_{m,n} U T \\ &= \widetilde{S}\widetilde{L} \cdot \widetilde{S}^{-1}(A + \Delta A)T^{-1} \cdot UT \\ &\equiv (I_m + E)^{-1}(A + \Delta A)(I_n + F)^{-1}. \end{aligned}$$

Working now to first order,

$$\widetilde{L} = I_m - [\,\mathrm{tril}(\widetilde{S}^{-1}\Delta A T^{-1})\ 0\,], \quad U = I_n - \mathrm{triu}(\widetilde{S}^{-1}\Delta A T^{-1}),$$

and so

$$E = \widetilde{S}(\widetilde{L}^{-1} - I_m)\widetilde{S}^{-1} = \widetilde{S}\,[\,\mathrm{tril}(\widetilde{S}^{-1}\Delta A T^{-1})\ 0\,]\,\widetilde{S}^{-1},$$

and similarly

$$F = T^{-1}(U^{-1} - I_n)T = T^{-1}\,\mathrm{triu}(\widetilde{S}^{-1}\Delta A T^{-1})T.$$

The result now follows on applying Theorem 3.2. □

In [4, Theorem 4.2], S and T are LU factors, and $\widetilde{S}_{12} = 0$, $\widetilde{S}_{22} = I_{m-n}$ is taken. For our application to QR factorization we take $\widetilde{S} = Q \in \mathbb{R}^{m\times m}$ and $T = R \in \mathbb{R}^{n\times n}$, and the expression for ϵ simplifies to

$$\epsilon = \|\,\mathrm{tril}(Q^{\mathrm{T}}\Delta A R^{-1})\| + \|R^{-1}\,\mathrm{triu}(Q^{\mathrm{T}}\Delta A R^{-1})R\|. \tag{3.2}$$

To evaluate ϵ we need to compute the backward error matrix $\Delta A = \widehat{Q}_1\widehat{R} - A$ explicitly, where $Q_1 = Q(:, 1:n)$, by forming the product $\widehat{Q}_1\widehat{R}$. But if we evaluate (3.2) using the computed ΔA then we will obtain an error estimate rather than a strict bound, because we will not have taken account of the rounding errors in evaluating ΔA (which is the result of massive cancellation). To obtain a strict bound we can either use

$$\Delta\widehat{A} = fl(\widehat{Q}\widehat{R} - A) = \Delta A + E, \quad |E| \leqslant \gamma_{n+1}(|\widehat{Q}||\widehat{R}| + |A|) \tag{3.3}$$

(where we assume that $\widehat{Q}$ has been formed explicitly) and bound ΔA by $|\Delta\widehat{A}| + \gamma_{n+1}(|\widehat{Q}||\widehat{R}| + |A|)$, or we can use a running or theoretical bound for $|\Delta A|$. In either case, we need to take absolute values and replace ϵ by an expression of the form

$$\|\,\mathrm{tril}(|Q^{\mathrm{T}}||\Delta A||R^{-1}|)\| + \|\,|R^{-1}|\,\mathrm{triu}(|Q^{\mathrm{T}}||\Delta A||R^{-1}|)|R|\,\|.$$

We note that it does not seem to be possible to evaluate $\mathrm{tril}(AB)$ or $\mathrm{triu}(AB)$ times a vector without forming the product AB explicitly, so condition estimation techniques do not seem to be applicable here.

4. Numerical experiments

To see how the practical behaviour of Algorithm SVD_QR compares with the theoretical predictions and to test the usefulness of the a posteriori bounds we have carried out numerical experiments very similar to those in [4, Section 4.1]. Our experiments were performed in MATLAB 5.2 on a Pentium workstation using simulated single precision: the result of every arithmetic operation is rounded to 24 bits, [3] so that $u = 2^{-24} \approx 5.96 \times 10^{-8}$.

[3] In fact, a few operations are not so-rounded, but we believe this makes no qualitative difference to the results.

We generated random matrices of the form $A = D_1 B D_2$, where $B = U\Sigma V^{\mathrm{T}}$ with U and V random orthogonal matrices and Σ a given matrix of singular values; B is constructed using the routine `rands\-vd` [4] from the Test Matrix Toolbox [8]. In each case the singular values are from an exponential distribution, $\sigma_i = \alpha^i$, and the condition number $\kappa(B) = 10^i$, $i = 1:7$. The matrices D_1 and D_2 are diagonal, with positive diagonal entries chosen from one of three pairs of random distributions: uniformly distributed logarithm (decreasing order for D_1 and increasing order for D_2); geometrically distributed entries (increasing order for D_1 and decreasing order for D_2); geometrically distributed entries in decreasing order for D_1 and uniformly distributed logarithm in increasing order for D_2. We took $\kappa(D_1) = \kappa(D_2) = 10^k$, $k = 2, 6, 12, 16$. The dimensions are $m = n = 16$ and one matrix of each type was generated. For each matrix we evaluated

$$\epsilon_i = \max\left\{ \frac{|\sigma_{\mathrm{D}}(k) - \sigma_{\mathrm{S}}(k)|}{\sigma_{\mathrm{D}}(k)} : A = D_1 B D_2,\ \kappa(B) = 10^i \text{ (12 such matrices)} \right\},$$

where $\sigma_{\mathrm{S}}(k)$ denotes the kth singular value computed in single precision by a particular algorithm and $\sigma_{\mathrm{D}}(k)$ denotes the kth singular value computed in double precision by Algorithm SVD_QR.

For Algorithm SVD_QR, provided that the growth factor $\rho_{m,n}$ and τ are both of order 1 then, in view of the inequalities (for $B = LU$)

$$\begin{aligned}\|B^{-1}\|^{1/2} &\leqslant \left(\|U^{-1}\|\|L^{-1}\|\right)^{1/2} \leqslant \max\left(\|U^{-1}\|, \|L^{-1}\|\right)\\ &= \max\left(\|B^{-1}L\|, \|UB^{-1}\|\right) \leqslant \|B^{-1}\| \max\left(\|L\|, \|U\|\right)\end{aligned}$$

from [4, Section 4.1], we expect (as a rather rough approximation) $\epsilon_i \approx \kappa(B)u \approx 10^{i-7}$, and the same approximation can be derived for Algorithm SVD with GECP [4, Section 4.1]. Table 1 reports the results for Algorithm SVD_QR with and without row sorting, Algorithm SVD with GECP, and the one-sided Jacobi algorithm; the "expected ϵ_i" column shows the approximation just described. The results for row pivoting were very similar to those for row sorting and hence are omitted. For the computation of τ and the backward error ΔB we tried the choices of D_1 and D_2 in (2.18) and (2.19), but report only the results for (2.19); the results for (2.18) were very similar.

The measured ϵ_i are of roughly the same order of magnitude as the expected values for Algorithm SVD_QR with row sorting. But when row sorting is omitted, the ϵ_i are much larger, as expected from the error analysis, since the growth factor $\rho_{m,n}$ is now unbounded. Algorithm SVD with GECP yields errors very similar to those from Algorithm SVD_QR with row sorting, as does the one-sided Jacobi algorithm. In their tests with random matrices, Demmel et al. [4, Section 4.1] also observe that the one-sided Jacobi algorithm gives similar accuracy to Algorithm SVD with GECP

[4] This routine is also accessible via the `gallery` function of MATLAB 5.

Table 1
Accuracy of SVD algorithms

i	Expected ϵ_i for Algorithm SVD_QR and Algorithm SVD with GECP	Measured ϵ_i for Algorithm SVD_QR with row sorting	Measured ϵ_i for Algorithm SVD_QR without row interchanges	Measured ϵ_i for Algorithm SVD with GECP	Measured ϵ_i for one-sided Jacobi SVD
1	10^{-6}	9×10^{-6}	1×10^{4}	1×10^{-5}	3×10^{-5}
2	10^{-5}	1×10^{-5}	1×10^{4}	6×10^{-6}	5×10^{-5}
3	10^{-4}	7×10^{-5}	5×10^{4}	6×10^{-5}	5×10^{-4}
4	10^{-3}	1×10^{-3}	6×10^{5}	6×10^{-4}	1×10^{-3}
5	10^{-2}	9×10^{-3}	1×10^{5}	6×10^{-3}	1×10^{-2}
6	10^{-1}	9×10^{-2}	9×10^{4}	5×10^{-2}	2×10^{0}
7	1	1×10^{0}	3×10^{5}	3×10^{-1}	1×10^{0}

and they give a matrix for which the one-sided Jacobi algorithm does not provide the desired relative accuracy

$$A(\gamma,\delta) = \begin{bmatrix} 1 & \gamma & \gamma \\ -\gamma & \gamma & \gamma^2 \\ 0 & \delta & 0 \end{bmatrix}, \tag{4.1}$$

where $0 < \delta \ll \gamma \ll 1$. For this matrix the elements determine the SVD to high relative accuracy, as can be shown using Theorem 3.1. For the matrix $A(10^{-6}, 10^{-12})$ we find that Algorithm SVD_QR with and without row sorting and Algorithm SVD with GECP all give computed singular values with relative errors of order u, while the one-sided Jacobi algorithm gives relative errors of order 10^{-2}.

These experiments show Algorithm SVD_QR with complete pivoting to perform just as well in practice as Algorithm SVD with GECP, and confirm that both can be superior to the one-sided Jacobi algorithm.

To check the relevance and sharpness of our error analysis for QR factorization we also computed, for $A = D_1 B D_2$ (assumed to be pre-pivoted) with D_1 and D_2 as in (2.19), the backward error matrix ΔB in $Q\widehat{R} = A + \Delta A = D_1(B + \Delta B)D_2$ and the bound for ΔB obtained by scaling (2.9) (ignoring the factor jm). The maxima corresponding to the matrices used for Table 1 are shown in Table 2. In every case the backward error $\|\Delta B\|/\|B\|$ is small with row sorting but very large without row interchanges and the bound is a good approximation. Concentrating now on row sorting, for the matrix C in Theorem 2.5, Table 3 reports the maximum values, for each i, of $\max_k \mu_k$ and $\rho_{m,n}$. The quantity ψ in (2.16) had maximum value over all the test matrices of 100 for D_2 defined by (2.18) and 3 for D_2 defined by (2.19). We conclude that, if we ignore the $f(m,n)$ term, Theorem 2.5 gives bounds that are within about three orders of magnitude of the actual backward error $\|\Delta B\|/\|B\|$ in these tests.

Table 2
Backward errors and bounds for QR factorization with column pivoting

i	With row sorting		Without row interchanges	
	$\|\Delta B\|/\|B\|$	Bound from (2.9)	$\|\Delta B\|/\|B\|$	Bound from (2.9)
1	4×10^{-6}	8×10^{-6}	3×10^{8}	4×10^{8}
2	7×10^{-6}	1×10^{-5}	1×10^{8}	1×10^{8}
3	4×10^{-7}	1×10^{-6}	6×10^{8}	2×10^{8}
4	1×10^{-6}	2×10^{-6}	2×10^{7}	1×10^{7}
5	3×10^{-7}	1×10^{-7}	5×10^{7}	3×10^{7}
6	3×10^{-7}	1×10^{-7}	1×10^{9}	7×10^{8}
7	2×10^{-7}	1×10^{-7}	2×10^{9}	2×10^{9}

Table 3
Values of μ_k in (2.15) and $\rho_{m,n}$ for C in Theorem 2.5

i	$\max_k \mu_k$	$\rho_{m,n}$
1	3×10^{1}	1×10^{2}
2	1×10^{2}	8×10^{1}
3	1×10^{2}	2×10^{1}
4	3×10^{2}	3×10^{1}
5	1×10^{3}	2×10^{0}
6	1×10^{2}	2×10^{0}
7	4×10^{3}	2×10^{0}

Table 4
Values of τ in (3.1)

i	QR with complete pivoting	GECP
1	1×10^{1}	4×10^{1}
2	5×10^{1}	2×10^{2}
3	2×10^{1}	2×10^{2}
4	6×10^{1}	2×10^{2}
5	1×10^{2}	1×10^{2}
6	9×10^{1}	1×10^{2}
7	9×10^{1}	9×10^{1}

The maximum values of τ for each i are shown in Table 4. These are all reasonably small, as expected.

Finally, we turn to a posteriori error bounds. We found the bounds based on Theorem 3.1 to be so pessimistic as to be useless; for each i no correct significant figures were predicted, with the bounds all at least 10^4. For the bound in Theorem 3.3 (which does not account for all the errors in the algorithm and is correct only to first order) we computed ΔA explicitly as a residual and included a term accounting for the errors in the evaluation of ΔA (see (3.3)); as shown by Table 5, the bound is at most four

Table 5
ϵ_i, with estimate and bound from Theorem 3.3

i	Algorithm SVD_QR with row sorting			Algorithm SVD with GECP		
	ϵ_i	Estimate	Bound	ϵ_i	Estimate	Bound
1	9×10^{-6}	2×10^{-5}	5×10^{-3}	1×10^{-5}	2×10^{-5}	6×10^{-3}
2	1×10^{-5}	2×10^{-5}	3×10^{-2}	6×10^{-6}	2×10^{-5}	4×10^{-2}
3	7×10^{-5}	2×10^{-4}	2×10^{-1}	6×10^{-5}	2×10^{-4}	1×10^{-1}
4	1×10^{-3}	5×10^{-3}	9×10^{-1}	6×10^{-4}	2×10^{-3}	1×10^{0}
5	9×10^{-3}	2×10^{-2}	8×10^{0}	6×10^{-3}	2×10^{-2}	1×10^{1}
6	9×10^{-2}	2×10^{-1}	7×10^{1}	5×10^{-2}	1×10^{-1}	2×10^{2}
7	1×10^{0}	1×10^{0}	9×10^{2}	3×10^{-1}	2×10^{0}	2×10^{3}

orders of magnitude larger than the actual relative error in every case for both QR factorization and GECP. The error estimate from Theorem 3.3 (which differs from the bound just mentioned in that it does not account for the errors in forming ΔA) is also shown in Table 5 and is seen to be a excellent estimate of the relative accuracy.

As a final test we tried the Kahan matrix

$$U_n(\theta) = \operatorname{diag}(1, s, \ldots, s^{n-1}) \begin{bmatrix} 1 & -c & -c & \ldots & -c \\ & 1 & -c & \ldots & -c \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & -c \\ & & & & 1 \end{bmatrix}, \tag{4.2}$$

where $c = \cos(\theta)$, $s = \sin(\theta)$. This matrix is well-known to cause difficulties for rank revealing factorizations because u_{nn} is of order 2^n times larger than the smallest singular value and yet $U_n(\theta)$ is invariant under QR factorization with column pivoting and GECP. We took $n = 24$, $\theta = 0.5$, and perturbed the diagonal slightly to ensure that rounding errors did not cause any pivoting to take place. As expected, Algorithm SVD_QR, Algorithm SVD with GECP and the one-sided Jacobi algorithm did not achieve high relative accuracy, since they work with poor RRDs: the maximum relative errors in the computed singular values were of order 10^{-3}, 10^{-4} and 10^{-3}, respectively. In this example the error estimate from Theorem 3.3 is of order 10^{-3} and the error bound of order 1 for both Algorithm SVD_QR and Algorithm SVD with GECP.

5. Conclusions

We have analysed the use of Householder QR factorization with complete pivoting for computing the RRD in Algorithm SVD. In implementing the algorithm we have to choose between two possibilities for the row interchanges: row sorting and

row pivoting. Row sorting guarantees that $\tau_1 \leqslant \sqrt{n}$ in (3.1), whereas τ_1 is unbounded for row pivoting; however, Theorem 2.5 is stronger for row pivoting than for row sorting. When implementing the QR factorization, after sorting the rows one can call any library routine for QR factorization with column pivoting, such as `xGEQPF` from LAPACK. Neither `xGEQPF` nor the code from [11] support row pivoting, and incorporating it into the latter code while retaining the use of level 3 BLAS is a nontrivial task. Therefore, given that the observed accuracy of Algorithm SVD_QR is similar for the two choices, we prefer row sorting.

We have shown that computing the explicit residual for QR factorization or GECP and using the bound from Theorem 3.3 gives useful a posteriori estimates and bounds for the relative error in the computed singular values.

Householder QR factorization with complete pivoting has several features that make it an attractive alternative to GECP in Algorithm SVD for computing the SVD of graded matrices.

1. In exact arithmetic, it provides just as good an RRD in the worst case and in practice, as measured by $\max(\kappa(X), \kappa(Y))$.
2. It reduces the cost of Algorithm SVD by the cost of an LU factorization with complete pivoting and a matrix multiplication.
3. Demmel [3] (see also [4, Algorithm 3.2]) proposes a generally more expensive version of Algorithm SVD containing an extra Jacobi SVD step in place of the QR factorization in step 2, and he states that for this version the factor $\kappa(\overline{R})$ in the bound of Theorem 1.1 can be removed. When the RRD is obtained from a QR factorization with complete pivoting Demmel's algorithm reduces to Algorithm SVD_QR, so the improved bound holds also for Algorithm SVD_QR. The $\kappa(\overline{R})$ term has not completely disappeared, however, because $\kappa(Y)$ is approximately equal to $\kappa(\overline{R})$.

In favour of GECP is that the conditions for the computed RRD to yield high relative accuracy in the SVD computed by Algorithm SVD are less restrictive than for QR factorization with complete pivoting. Moreover, Algorithm SVD with GECP is more versatile than we have indicated. Provided that GECP is implemented in an appropriate nonstandard fashion in each case it can be used to compute the SVD to high relative accuracy for various classes of structured matrices, including total signed compound matrices (which include acyclic matrices), Cauchy matrices, totally positive matrices, and Vandermonde matrices [3,4]. It is an open problem how to obtain accurate SVDs of such matrices using QR factorization with complete pivoting.

Acknowledgements

I benefited from a stimulating discussion on this work with the authors of [4] at the International Workshop on Accurate Solution of Eigenvalue Problems, Pennsylvania State University, July 1998. I am grateful to Jim Demmel and Zlatko Drmač for their helpful comments on the manuscript.

References

[1] E. Anderson, Z. Bai, C.H. Bischof, J.W. Demmel, J.J. Dongarra, J.J. Du Croz, A. Greenbaum, S.J. Hammarling, A. McKenney, S. Ostrouchov, D.C. Sorensen. LAPACK Users' Guide, Release 2.0, second ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1995, xix+325 pp. ISBN 0-89871-345-5.

[2] A.J. Cox, N.J. Higham, Stability of Householder QR factorization for weighted least squares problems, in: D.F. Griffiths, D.J. Higham, G.A. Watson (Eds.), Numerical Analysis 1997, Proceedings of the 17th Dundee Biennial Conference, Pitman Research Notes in Mathematics, vol. 380, Addison-Wesley, Longman, Harlow, Essex, UK, 1998, pp. 57–73.

[3] J.W. Demmel, Accurate SVDs of structured matrices, Technical Report CS-97-375, Department of Computer Science, University of Tennessee, Knoxville, TN, USA, October 1997, 19 pp. LAPACK Working Note 130, SIAM J. Matrix Anal. Appl., to appear.

[4] J.W. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić, Z. Drmač, Computing the singular value decomposition with high relative accuracy, Linear Algebra Appl. 299 (1999) 21–80.

[5] J.J. Dongarra, J.R. Bunch, C.B. Moler, G.W. Stewart, LINPACK Users' Guide, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1979, ISBN 0-89871-172-X.

[6] Z. Drmač, On principal angles between subspaces of Euclidean space, Technical Report CU-CS-838-97, Department of Computer Science, University of Colorado at Boulder, March 1997.

[7] S.C. Eisenstat, I.C.F. Ipsen, Relative perturbation techniques for singular value problems. SIAM J. Numer. Anal. 32 (6) (1995) 1972–1988.

[8] N.J. Higham, The test matrix toolbox for Matlab (version 3.0). Numerical Analysis Report No. 276, Manchester Centre for Computational Mathematics, Manchester, England, September 1995, p. 70.

[9] N.J. Higham, Accuracy and Stability of Numerical Algorithms. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1996. xxviii+688 pp. ISBN 0-89871-355-2.

[10] M.J.D. Powell, J.K. Reid, On applying Householder transformations to linear least squares problems, in: Proceedings of the IFIP Congress 1968, North-Holland, Amsterdam, The Netherlands, 1969, pages 122–126.

[11] G. Quintana-Ortí, X. Sun, C.H. Bischof, A BLAS-3 version of the QR factorization with column pivoting, SIAM J. Sci. Comput. 19 (5) (1998) 1486–1494.